第7章

AC '97音声出力回路の製作

——AC-linkインターフェースでCODEC LSIを制御する

井倉将実

ここでは，本誌付属のFPGA基板を使った音声出力回路の製作例を紹介する．音声CODECとして広く使われているAC (Audio Codec) '97に対応する．AC '97 CODEC LSIを，AC '97で規格化されたバスによって制御し，音声出力を行う回路を設計した．また，AC '97規格の概要についても解説する．（編集部）

本稿では，本誌付属のFPGA基板を使い，AC '97 CODEC LSIを制御する回路の設計例を紹介します(**写真1**)．また，AC '97 CODEC LSIを使い，2チャネルのステレオ音声出力も行います．

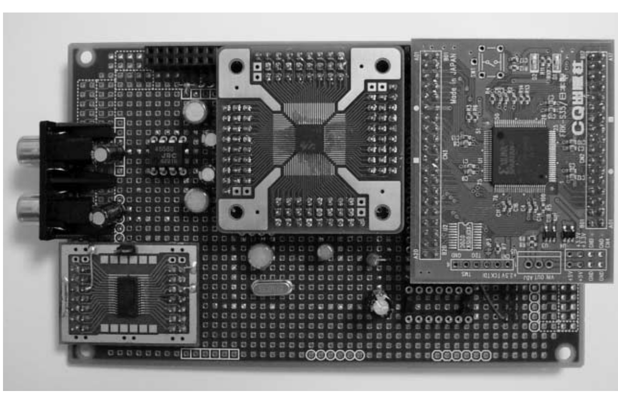


写真1　AC '97音声出力回路の外観
AC-linkインターフェースでCODEC LSIを制御して音声出力を行う．今回は，ホスト・マイコンとしてSH-4を使用した．

1．音声CODEC AC '97とは，

音声出力用D-Aコンバータのインターフェースには，一般的なパラレル・バス以外に，メーカ独自のバス規格が使われることがあります．例えば，米国Texas Instruments社の「PCM1716E」ファミリは，カー・オーディオやセットトップ・ボックスに使われている96kHz，24ビットの音声用D-Aコンバータですが，バス・インターフェースとしてI^2S (Inter-IC Sound)バスと呼ばれる3線式シリアル方式が用いられています．これは，クロック(SCK)，シリアル・データ(SD)，ワード選択(WS，現在のデータがスピーカの左右どちらのものであるかを示す)で制御されます(**図1**)．

I^2Sバスは，8ビット・マイコンでも容易に扱える単純なインターフェースであるため，非常に幅広く使用されています．ただし，2スピーカ向けであり，昨今流行の5.1サラウンド・スピーカには対応していません．また，双方向バスではないので，A-Dコンバータと D-Aコンバータを同一バスで駆動することができません．

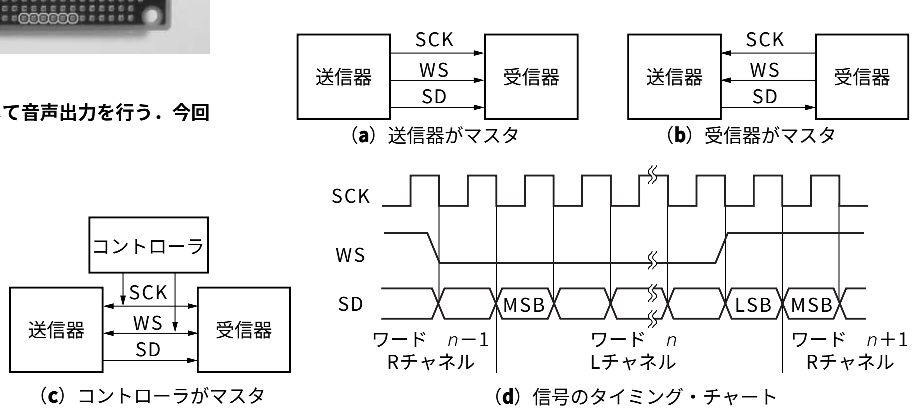


図1
I^2Sバスの概要[1]
クロック(SCK)，シリアル・データ(SD)，ワード選択(WS)の3本の信号で制御する．

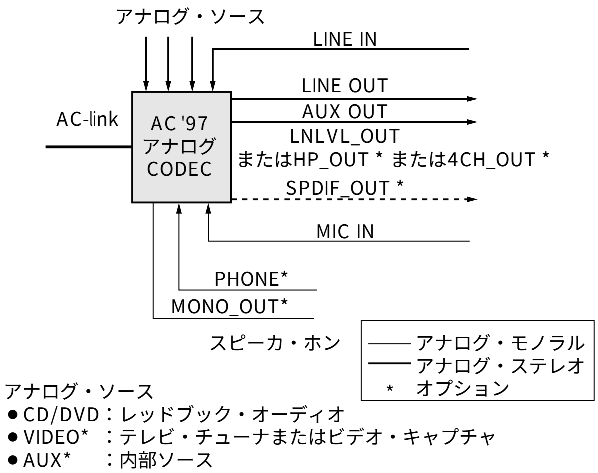


図2 Audio Codec ’97（AC ’97）の概要(2)

米国Intel社が提唱した，多チャネル・スピーカやマイク/ライン入力などを包括して取り扱うことのできる音声CODEC規格．

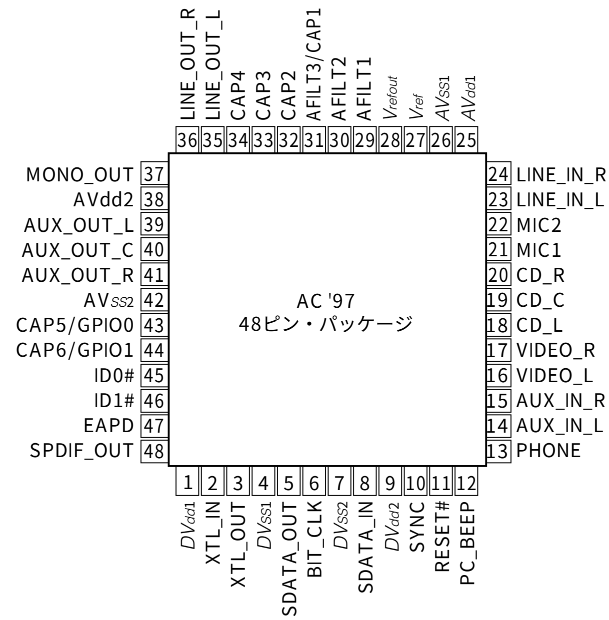


図3 AC ’97のピン配置やパッケージに関する指針(2)

AC ’97は，バスの信号線だけでなく，LSIを設計する際のピン配置やパッケージに関する指針，電気的特性なども含む規格である．

現在のマルチメディア機器は，2スピーカでは心もとない状況といえます．そこで米国Intel社は，多チャネル・スピーカやマイク/ライン入力などを包括的に取り扱える新しい音声CODEC規格を提唱しました．それが「Audio Codec ’97（AC ’97）Component Specification」です（**図2**）．現在は，2002年4月にリリースされたRevision 2.3が最新版です．

2. AC ’97とAC-link

AC ’97は，バスの信号線だけでなく，LSIを設計する際のピン配置やパッケージに関する指針，電気的特性なども含む規格です（**図3**）．

AC ’97規格準拠のLSIの基本ブロックは，以下のとおりです．

- 音声出力ブロック
- 音声入力ブロック
- 音声ミキシング・ブロック
- AC-link制御ブロック

このほか，外部I/Oアクセスを行うブロックやクロック生成部があります．

AC-linkを持つLSIが，これらすべてのブロックを実装しているとは限りません．例えば，英国Wolfson Microelectronics社の「WM9709」は，AC-link制御ブロックと音声出力ブロックしか実装していません．このLSIは，20ビット，2チャネルのステレオ・オーディオLSIとして使用されることを想定した小型・低消費電力のLSIです．従来のI^{2}Sバスが AC-linkに変わっただけのような仕様になっています．MP3プレーヤのように音声出力しか想定していない機器であれば，これでまったく問題ありません．

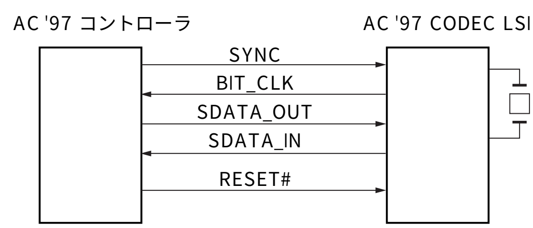


図4 AC-link(2)

AC ’97のバスは，5本の信号線でアクセスする．このバスを「AC-link」と言う．BIT_CLKは，図ではAC ’97 CODEC LSIから出力されているが，AC ’97コントローラから出力されることもある．

● 5本の信号線でアクセスするAC-link

AC ’97のバスには，**図4**のように5本の信号線でアクセスします．このバスを「AC-link」と言います．AC ’97規格に対応したLSIは，AC-link接続によってマイコンなどから制御されます．

AC-linkを構成する5本の信号線の意味を**表1**にまとめます．RESET#，SYNC，SDATA_OUTは，AC ’97準拠

のLSIを制御するコントローラから見て出力になります．SDATA_INはコントローラから見ると入力になります．

● AC-link vs. I²S

従来のI²S信号規格では，2チャネルの音声しか出力できませんでした．これに対して，AC-linkは音声出力以外の用途にも使うことができます．AC '97準拠LSIにA-Dコンバータなどの機能があれば，入力信号をサンプリングしてそれを取り込めます．AC '97準拠LSIは，アナログ・アンプ回路の出力イネーブル機能やA-D/D-Aコンバータの制御レジスタを持ちます．そこで，例えばLチャネルだけを出力し，Rチャネルはミュート（出力抑制）することが可能です．

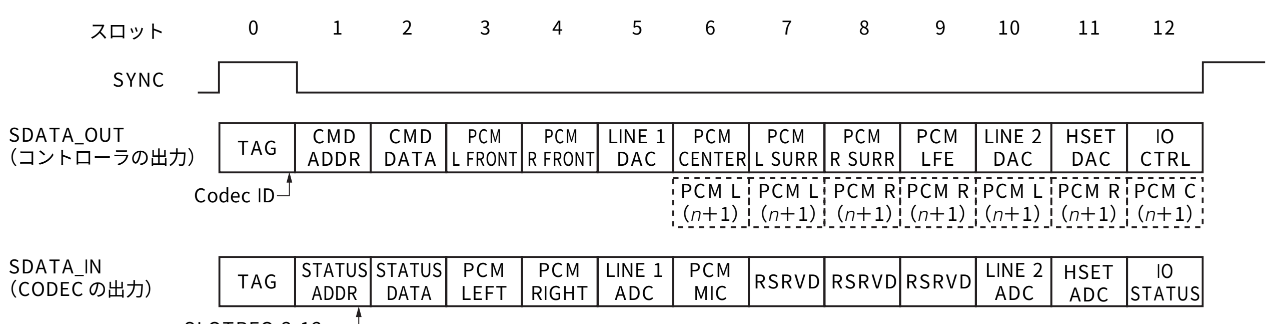
このようにAC-linkとI²Sの信号を比較してみると，**表2**に示すように多くの点で機能が改善されていることがわかります．ただし，AC '97規格では最大データ長とサンプリング・レートが決まってしまっているため，駆動チャネル数（スピーカ数）が多いとはいえ，音質という点だけを見ればI²Sに負けています．実際に聞き比べてみると，このあたりはI²Sに分があります．一方，マイク，ビープ，ライン入力と出力をミキシングして出力できること，また，AC-link経由でデバイス内部のレジスタにアクセスしてボリュームやアッテネータなどを調節できること，最大でステレオ6チャネルを実現できることについては，AC '97の

表1　AC-linkを構成する信号線

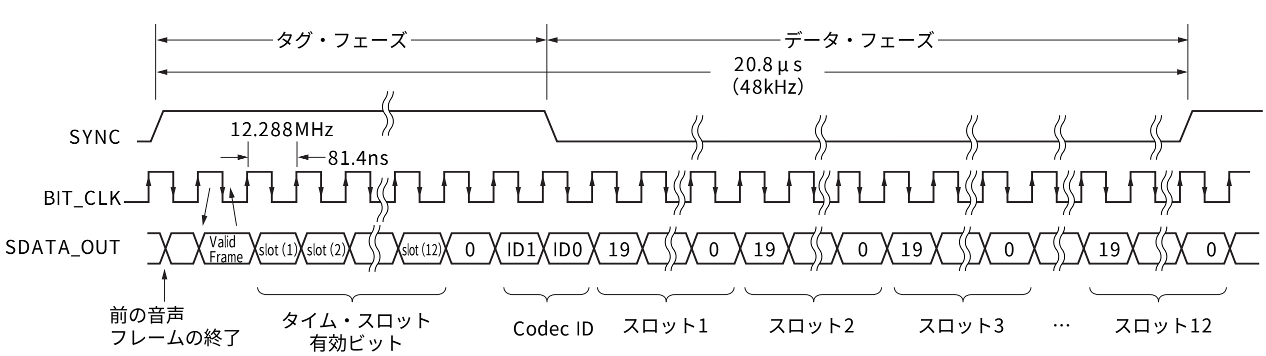
信号名	論　理	コントローラから見た方向	意　味
RESET #	負論理	出力	ハードウェア・リセット
BIT_CLK	正論理	入力または出力	AC-linkのバス・クロック
SYNC	正論理	出力	AC-linkのスロット番号0を示す同期信号
SDATA_IN	正論理	入力	シリアル・データ入力
SDATA_OUT	正論理	出力	シリアル・データ出力

表2　AC-linkとI²Sの信号を比較

	AC-link	I²S
転送方向	双方向	片方向
信号本数	5線式シリアル	3線式シリアル
データ・ビット長	256ビット固定	32～64ビット
最大データ長	20ビット	24ビット
サンプリング・レート	48kHz	32kHz～192kHz
分解能	20ビットPCM	24ビットPCM
駆動チャネル数	6チャネル	2チャネル
ミキシング機能	あり	なし



（a）転送方式



（b）データ転送の詳細

図5　AC-linkの動作(2)

スロット0からスロット12までの合計13個のスロットを並べてデータ転送を行う．タグ・フェーズのValid FrameからID0までがスロット0のデータである．

ほうが有利です．

● AC-link転送の基本単位は「スロット」

AC-linkは，「スロット」と呼ばれるデータのビット列の集まりを最小の単位とします．スロット0からスロット12までの合計13個のスロットを並べてデータ転送を行います（**図5**）．

先頭のスロット0は特別な意味を持ちます．スロット語長は16ビット固定長です．また，スロット0の転送中にSYNC信号を“H”レベルにすることで，シリアル・ライン上のビットストリームが先頭であることを知ることができます．スロット0は「タグ・フェーズ（Tag Phase）」と呼びます．

スロット1からスロット12までは「データ・フェーズ（Data Phase）」と呼びます．各スロットのスロット語長は20ビットです．

したがって，タグ・フェーズの16ビットと，データ・フェーズの20ビット×12スロットの合計256ビットが1回のAC-linkオペレーションとなります．これを「フレーム」と言います．48kHzサンプルの256倍がビット・クロックになるので，ビット・クロックは12.288MHzと規定されています．

表3（a）は，AC ′97 CODEC LSIの内部レジスタへの書き込みとD-Aコンバータへ出力するSDATA_OUTのスロットの意味を表したものです．また**表3（b）**は，内部レジスタのステータス表示とA-Dコンバータによって取り込まれるSDATA_INのスロットの意味を表したものです．

スロット0には，スロット1からスロット12までの各スロット内のデータが有効（＝1）か無効（＝0）かを示すビット列が設定されます（**表4**）．かりにステレオ2チャネルのオーディオ出力しか使わない場合，スロット3が有効であることを示すビット12と，スロット4が有効であることを示すビット11に‘1’をセットすれば，AC-linkオペレーション1回分で2チャネル分のステレオ出力を行えるというわけです．オーディオ取り込みモードにデバイスが設定されており，かりにビット12とビット13が‘1’に設定されていれば，A-D変換後の音声データ・ストリームは有効です．

表3 スロットの意味[(2)]

スロット	スロット名称	説明
0	SDATA_OUT TAG	書き込みスロット有効タグ
1	Control CMD ADDR write port	AC-linkコマンド・アドレス書き込みポート
2	Control DATA write port	16ビット・コマンド・データ書き込みポート
3，4	PCM L&R DAC playback	16，18，20ビットPCMデータ，L＆Rチャネル
5	Modem Line 1 DAC	16ビット・モデム用ライン1
6，7，8，9	PCM Center，Surround L&R，LFE	16，18，20ビットPCMデータ，センタ，サラウンドL＆R，リア
10	Modem Line 2 DAC	16ビット・モデム用ライン2
11	Modem handset DAC	16ビット・モデム用ライン3またはハンドセット
12	Modem IO control	モデム機器用汎用I/Oバス・コントロール（コントロール書き出しなど）
10，11	SPDIF Out	SPDIF出力系統（オプション）
6～12	Double rate audio	88.2k/96kbps倍速レート・オーディオ・スロット（オプション）

（**a**）AC-link出力スロット

スロット	スロット名称	説明
0	SDATA_IN TAG	読み出しスロット有効タグ
1	STATUS ADDR read port	AC-linkコマンド・アドレス読み出しポート（書き込みスロット1の写し）
2	STATUS DATA read port	16ビット・コマンド・データ読み出しポート
3，4	PCM L&R ADC record	16，18，20ビットPCMデータ，L＆Rチャネル
5	Modem Line 1 ADC	16ビット・モデム用ライン1
6	Dedicated Microphone ADC	16，18，20ビットPCMデータ，センタ，サラウンドL＆R，リア
7，8，9	Vendor reserved	未定義，ベンダ機能拡張用
10	Modem Line 2 ADC	16ビット・モデム用ライン2
11	Modem handset input ADC	16ビット・モデム用ライン3またはハンドセット
12	Modem IO status	モデム機器用汎用I/Oバス・コントロール（ステータス読み出しなど）

（**b**）AC-link入力スロット

このデータをメモリやハード・ディスク装置などに取り込んでおけば，ROW-PCMデータとして使用できます．

● 多くのメーカがAC '97 CODEC LSIを出荷

AC '97規格に準拠したLSIは，米国Analog Devices社や台湾のC-media社など，多くのメーカが出荷しています．

筆者は，Wolfson Microelectronics社の低消費電力AC '97 CODEC LSIである「WM9711」をよく使います．**図6**にこのWM9711の概要を示します．

この製品の最大の特徴は，ほかの同等製品と比べて低消費電力で動作することです．また，パッケージとしてAC '97仕様で紹介されている48ピンTQFP以外に，7mm角の48ピンQFNが用意されています．アンプが内蔵されており，ヘッドホンや小型スピーカの駆動も可能です．さらに，ステレオのライン入力やマイクなど，最大4系統の入力ソースに対するA-D変換機能や音声出力とのミキシング機能を持ちます．例えば，1チップでカラオケ・マシンのようなアプリケーションを実現できます．

このほか，二つのコンパレータを内蔵しており，バッテリ・アラーム機能も実現できるようです．ただし，筆者はこの機能はまだ使ったことがありません．

動作電圧範囲は1.8V～3.6Vと使いやすくなっています．機能的には十分すぎる感がありますが，いらない機能は内蔵レジスタでOFFすることもできます．

表4　スロット0の詳細(2)

ビット	機　能
15	フレーム有効
14	スロット1：コマンド・アドレス・ビット有効（プライマリCODECのみ）
13	スロット2：コマンド・データ・ビット有効（プライマリCODECのみ）
12	スロット3：LチャネルPCMのデータ有効
11	スロット4：RチャネルPCMのデータ有効
10	スロット5：モデム・ライン1のデータ有効
9	スロット6：センタ・チャネルPCMのデータ有効
8	スロット7：LチャネルPCMサラウンドのデータ有効
7	スロット8：RチャネルPCMサラウンドのデータ有効
6	スロット9：PCM LFEのデータ有効
5	スロット10：モデム・ライン2またはLチャネルPCM（n＋1）のデータ有効
4	スロット11：モデム・ハンドセットまたはRチャネルPCM（n＋1）のデータ有効
3	スロット12：モデム汎用I/Oまたはセンタ・チャネルPCM（n＋1）のデータ有効
2	予約
1～0	2ビットのCodec ID

3. AC-linkコントローラの設計

付属FPGA基板を使ってAC '97 CODEC LSIを制御し，音声を出力する回路を設計します．ここではAC-linkインターフェースを使って制御します．CODEC LSIとしては，WN9711を使用します．**図7**に回路図を示します．

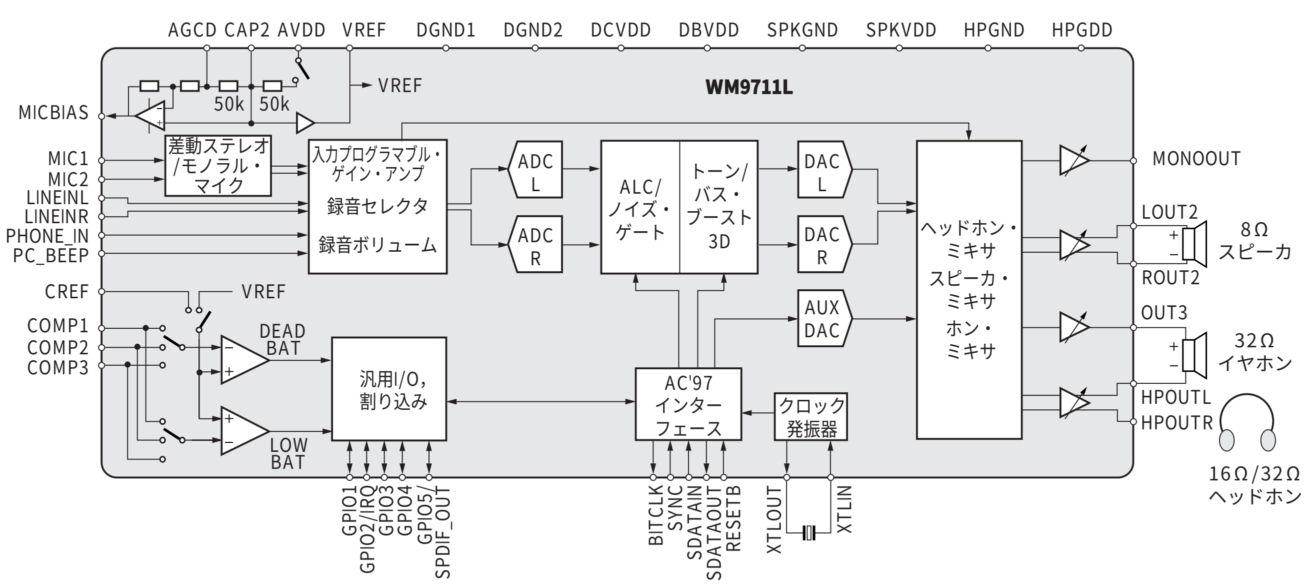


図6　WM9711のブロック図(3)

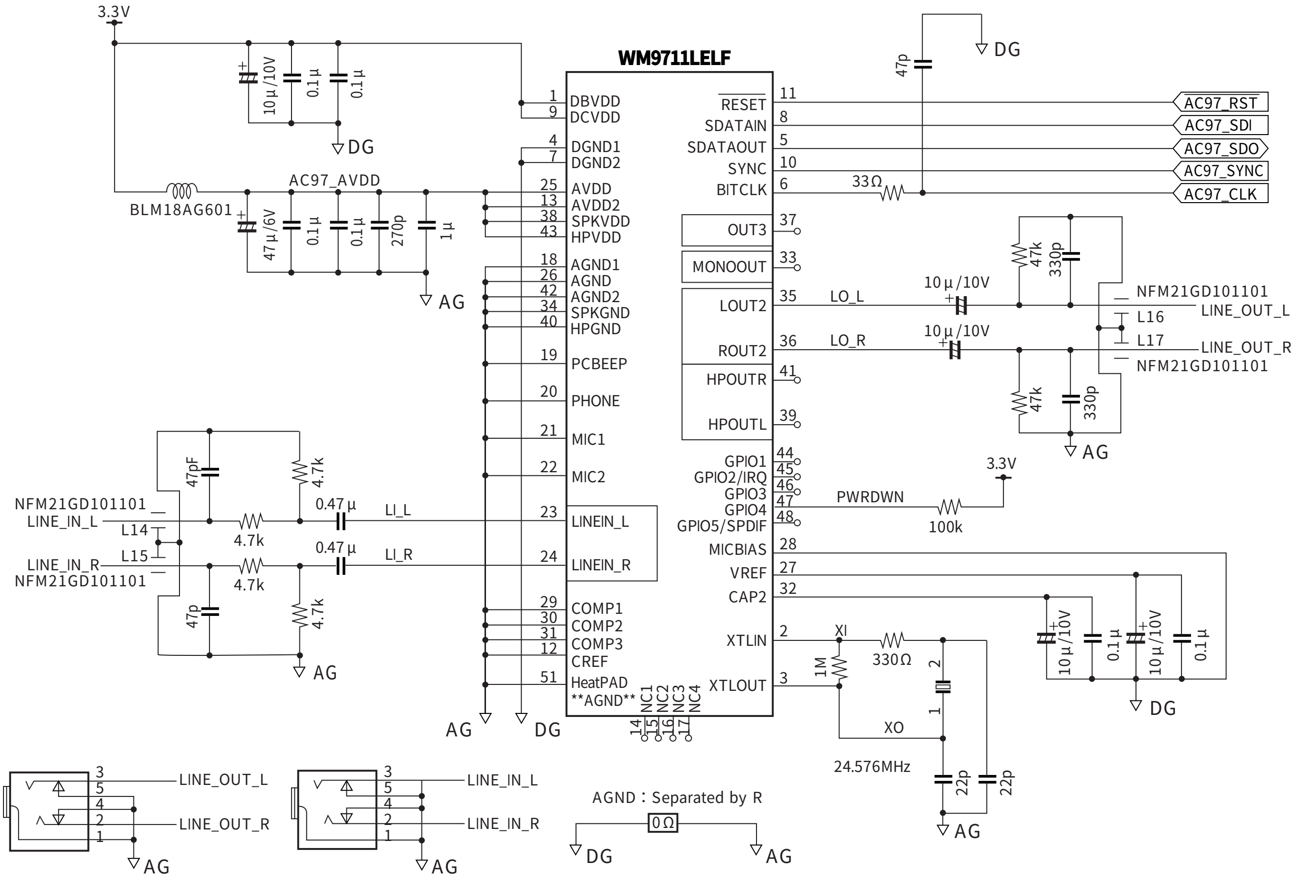


図7　AC '97音声出力回路の回路図

WM9711のBITCLKは，レジスタ設定により入力信号としても使用できる．今回は，コントローラから出力するように設計した．

コラム　AC '97 CODEC LSIのレジスタについて

AC '97 CODEC LSIで音声を出力する場合，LSIのレジスタにアクセスしなければなりません．

レジスタは0x00番地から0x7F番地までの128バイトです．AC '97規格で並びや機能が規定されているため，一度設計すれば，すべてのAC '97 CODEC LSIを動作させることができます．

プロセッサ側では，ミュート（オーディオ出力の抑制）機能を解除したり，電子ボリュームを設定したりする必要があります．

外部プロセッサなどからCODEC LSI内のレジスタ空間にアクセスするには，いくつかの実現方法があると思います．本稿で設計している回路は，以下の方法で内部レジスタを仮想的に表現しています．

- アドレス幅6ビット（64ワード分）の16ビット・データ長レジスタをFPGA内部にイメージしている．
- 外部プロセッサからいずれかの空間に対する書き込み要求が行われた場合，そのアドレスと書き込みデータをAC-link制御回路に引き渡して，適当な時期のフレームでAC '97 CODEC LSIに書き込む．このときのスロット1のビット19には書き込みを示す'0'がセットされ，2倍されたアドレスがセットされる．スロット2には書き込みデータをセットする．そして，スロット0のビット14～ビット13には'1'をセットし，スロット1とスロット2が有効であることを示す．
- 外部プロセッサからの内蔵レジスタの読み出し要求では，スロット1のビット19には読み出しを示す'1'をセットし，スロット2はすべて'0'（実はダミー）を設定する．スロット0のビット14～ビット13には'1'をセットし，スロット1とスロット2が有効であることを示す．これで一度，AC-linkのフレームが発行される．次のフレームでSDATA_INを監視し，スロット2のデータを読み取る．つまり，読み出しには1フレームの遅れが発生する．読み出されたデータは，別に用意する読み出しレジスタに保持する．
- 書き込み時，読み出し時ともにAC-link制御回路のステータス・レジスタで，現在の状況を表す．1フレームが48kHz単位であるため，プロセッサの動きから見ると非常に低速であり，ソフトウェアはステータス・レジスタ内のこれらのフラグの動きを見ながらレジスタ・アクセスを行う必要がある．

表5　各スロットのビット長とビットの意味

スロットの有効/無効を表す制御用スロット（16ビット長）	
ビット15	フレームが有効であるかどうかを指定．‘1’で有効
ビット14	スロット番号1（アドレス指定スロット）が有効であるかどうかを指定
ビット13	スロット番号2（データ設定スロット）が有効であるかどうかを指定
ビット12	スロット番号3（Lチャネル PCM音声出力スロット）が有効であるかどうかを指定
ビット11	スロット番号4（Rチャネル PCM音声出力スロット）が有効であるかどうかを指定
ビット10～ビット0	スロット番号5～スロット番号12まで未使用であるため，‘0’を設定

（a）スロット番号0

WM9711内蔵レジスタのアクセス・アドレス指定スロット（20ビット長）	
ビット19	データ転送方向指示．‘1’＝読み出し，‘0’＝書き込み．今回はレジスタの書き込みのみ
ビット18～ビット12	内蔵レジスタのアクセス・アドレスを指定する．データ語長が16ビットなので，実際にはビット12は常時‘0’となる
ビット11～ビット0	将来的な機能拡張部分．すべて‘0’を設定

（b）スロット番号1

WM9711内蔵レジスタのデータ・レジスタ（20ビット長）	
ビット19～ビット4	内部レジスタ・アクセス用の16ビット語長データ
ビット3～ビット0	将来的な機能拡張部分．すべて‘0’を設定

（c）スロット番号2

Lチャネル PCM音声出力スロット（20ビット長）	
ビット18～ビット2	18ビットLチャネル側 PCMデータ設定用

（d）スロット番号3

Rチャネル PCM音声出力スロット（20ビット長）	
ビット18～ビット2	18ビットRチャネル側 PCMデータ設定用

（e）スロット番号4

表6　コントロール・レジスタ

アドレス	レジスタ名称	ビット															
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
$1000	ACLINKアドレス										ADRS［6：0］						
$1002	ACLINKライト・データ・バッファ	DATA［15：0］															
$1004	ACLINKリード・データ・バッファ	DATA［15：0］（Reserved）															
—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
$1008	コマンド・レジスタ	ACEN						FLEN	FREN							REN	WEN
$100A	ステータス・レジスタ	ACACT						FLACT	FRACT	RLOVF	RROVF	FLOVF	FROVF	RDACT	WRACT	RDCMP	WRCMP
$100C	割り込み制御レジスタ									RLOVI	RROVI	FLOVI	FROVI			RDCI	WRCI
—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
$1010	Lチャネル・バッファ・ポインタ							LBufPTR［9：0］＝0～1023									
$1012	Rチャネル・バッファ・ポインタ							RBufPTR［9：0］＝0～1023									

- コマンド・レジスタ
  - ACEN：AC-link転送と内部コントローラの動作を有効にする
  - FLEN：前方・Lチャネル PCM転送を有効にする
  - FREN：前方・Rチャネル PCM転送を有効にする
  - REN　：AC-linkの読み出し処理要求を行う
  - WEN　：AC-linkの書き込み処理要求を行う
- 割り込み制御レジスタ
  - ［RL/RR/FL/FR］OVI：各チャネルのポインタ・オーバフローが発生した場合の割り込み発行を許可する
  - RDCI　：AC-linkの読み出し処理が完了した際の割り込み発行を許可する
  - WRCI　：AC-linkの書き込み処理が完了した際の割り込み発行を許可する
- バッファ・ポインタ
  - LBufPTR/RBufPTR：Lチャネル/Rチャネル・バッファ読み出しポインタ・アドレス
- ステータス・レジスタ
  - ACACT　：AC-linkの内部シーケンサが動作している最中であることを示す
  - FLACT　：前方・Lチャネル PCM転送が有効であることを示す
  - FRACT　：前方・Rチャネル PCM転送が有効であることを示す
  - ［RL/RR/FL/FR］OVF：各チャネルのポインタ・オーバフローが発生したことを示す
  - WRACT　：AC-linkの書き込み処理が行われている最中であることを示す
  - WRCMP　：AC-linkの書き込み処理が完了したことを示す

表7　メモリ・マップ

先頭4KバイトはLとRのPCMバッファを交互に配置している（32ビットCPUでL/R一括書き込みを考慮したため）．

アドレス	31-24	23-16	15-08	07-00
	+$03	+$02	+$01	+$00
$0000	L［0］		R［0］	
$0004	L［1］		R［1］	
$0008	L［2］		R［2］	
・	・		・	
・	・		・	
・	・		・	
・	・		・	
$0FF4	L［1021］		R［1021］	
$0FF8	L［1022］		R［1022］	
$0FFC	L［1023］		R［1023］	
$1000	ACLINK ライト・データ		ACLINKアドレス	
$1004			ACLINK リード・データ	
$1008	ステータス・レジスタ		コマンド・レジスタ	
$100C			割り込み制御レジスタ	
$1010	Rチャネル・バッファ・ポインタ		Lチャネル・バッファ・ポインタ	
・	・		・	
・	・		・	
・	・		・	
・	・		・	
$1FFC				

● 2チャネルの音声出力のみ実現

今回の設計では，WM9711の全機能は使用しません．2チャネルのステレオ音声出力のみ実現します．

必要なAC-linkスロットは，**表3**を参考にすると，スロット0からスロット4までの5スロットとなります．各スロットのビット長とビットの意味を**表5**に示します．また，SDATA_OUTピンしか使わないので，WM9711は音声出力のみを行います．

なお，スロット0については，今回設計するハードウェア側で以下のような制御が行われます．

- ビット15，ビット12，ビット11は，AC-link制御回路内に実装されているコントロール・レジスタ(**表6**)内のフレーム開始フラグとL/RのPCMデータ出力開始フラグの値が反映される．これらの3フラグのすべてが1セットである場合は，フレームとL/Rチャネルの両方が有効になる．
- ビット14とビット15は，AC-link制御回路内に用意されたAC '97レジスタ・アクセス空間をアクセスされたときだけ有効になる(p.81のコラム「AC '97 CODEC LSIのレジスタについて」を参照)．

スロット3とスロット4に出力されるPCMデータは，AC-link制御回路内のポインタ・レジスタによって参照された2組のFPGA内蔵メモリ・ブロック(Block RAM)から取り出したデータが設定されます．このポインタは，コントロール・レジスタ内のPCMデータ出力開始フラグが'1'にセットされると，フレームが完了するたびに1アドレスずつ加算されます．そのため，アドレスの動きとしては0番地から1023番地まで移り，そのフレームが終わると0番地にオーバラップします(**表7**)．この動作はPCMデータ出力開始フラグが0リセットされるまで繰り返されます．

また，この開始フラグは，スロット番号0のビット12とビット11にも適用されます．

● Spartan-3の構造の特徴を生かす

今回は，付属FPGA基板に実装されたSpartan-3の特徴を生かした設計を考えます．

XC3S50は，四つの18Kビットの内蔵メモリ・ブロック(Block RAM)を搭載しています．このBlock RAMは18ビット×1,024ワード分のメモリとして使用することができます．1個のBlock RAMで利用できる最大データ・バス幅はちょうどCDクオリティの音声PCMデータ語長(16ビット)と一致しており，1個のBlock RAMで1チャネル分のPCMデータ保持バッファを実現できます．そこで，このバッファを2個使って，2チャネル・ステレオのL/Rバッファを実現することにします．

実は過去に，PCIバス上にI^2Sインターフェースを持つPCM対応D-Aコンバータを使って，Windows 2000搭載のパソコンに接続して利用するサウンド・カードを作ったことがあります．このとき，検証サンプルとしてハード・ディスク上に用意されたPCMデータを48kHzで定期的にPCIバスへ出力しようとしたところ，音切れを起こすことが多々ありました．48kHzからすると800倍程度も早いPCIバスとはいえWindows自身はリアルタイムOSではないので，定期的な転送要求に対する即座の応答を保証できません．そのためデータ転送が間に合わず，音切れが発生したというわけです．

それではどれくらいのバッファ・サイズがあると48kHzサンプルのPCMを確実に出力できるのか，Pentium III(1GHz)とi815チップセットが実装された当時最速のメイン・ボードで調べたところ，おおむね1Kワード分のキュー(ここではバッファと同義)を用意すれば，よほどの操作を行わない限り音が飛ばないだろうという結論に到達しました．つまり21ms分(1/48000×1024)です．音楽CDのように音データを垂れ流しにするアプリケーションではバッファ容量をもっと増やしても良いのですが，あまりに多すぎると今度はPCIバス上の転送量が増えるとともに，ゲームのようなアプリケーションでは効果音を出すタイミングが遅くなり，画面と音が同期しなくなります．

そこで，実現するFPGAの容量を考えると1K～4Kワード程度のキューでよいと思われます．

図8は今回設計したAC '97 CODECデバイス制御回路のブロック図になります．ステレオ音声データはCPUなどの外部コントローラからバス・インターフェースを経由してFPGA内部のBlock RAMに転送されます．このメモリ・バッファ容量は今回は1Kワードになります．さらにこのバッファをデュアルポート・メモリ構成にすることで，外部コントローラからのPCMデータの充てんと，AC-linkのスロット3とスロット4に対するデータ充てん用の読み出しが非同期に行えます．これもSpartan-3のBlock RAMのおもしろい使いかたのひ一つであり，中小規模のCPLDでは実現が難しいところです．

AC-linkを制御しているシーケンサでは，ステータス/コ

7

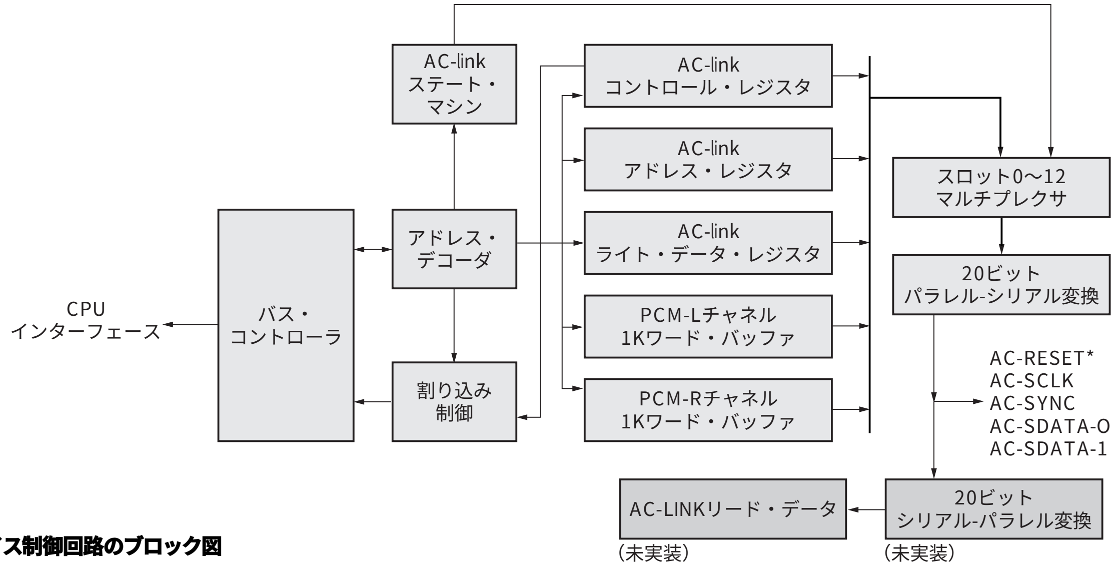


図8
AC ʼ97 CODECデバイス制御回路のブロック図

ントロール・レジスタの内容に応じて，AC-link用の20ビット・シフト・レジスタにタイミング良くスロット番号0からスロット番号12までを挿入して，パラレル-シリアル変換を行います．このシーケンサでは，48kHz，つまり256ビットのデータ・ストリームが発行されるたびにスロット番号0からの処理に戻って，再度データの充てんを行います．注意点としては，スロット0の有効ビットはステータス/コントロール・レジスタ内の値がそのまま反映されることです．

このほか，一度起動されたAC-linkオペレーションを，外部コントローラのつごうにより強制的に終了する場合もあります．そのような場合，AC-linkオペレーション中に(例えば，スロット番号0のデータ・ストリーム出力中に)動作を停止させるわけにはいかないので，ステート・マシンの開始時点と終了時点でAC-link有効フラグをチェックする機能を入れています．これにより，いったん開始されたAC-linkオペレーション中に，動作が有効であればスロット0からの処理に戻り，停止状態であればスロット12の処理を行った後にアイドル・ステートに戻って，再度AC-linkオペレーションの開始を待つこととします．

AC ʼ97 CODEC LSI内の制御レジスタへのアクセスが行われた場合，ステート・マシンはスロット0，スロット1，スロット2に要求された値を1回だけ挿入するような処理を行います．もちろん，現在のAC-linkオペレーションでPCMデータの出力と同時に行われる場合もあります．レジスタへのアクセス処理がステート・マシンで行われている間は，ステータス・レジスタ上に「レジスタ・アクセス進行中」のステータス・フラグをセットするようにしています．このフラグを適宜監視することで，外部コントローラ側では，要求したレジスタ・アクセスの進行状態がわかります．

とくに，1レジスタ・アクセスが48kHzという，CPUのアクセスに比べれば非常にゆっくりとした速度で実行されます．ステータス・フラグの変化待ちをチェックするポーリング処理では，CPUはほとんど停止している状態になるでしょう．そこで，高度に設計するのであれば，割り込み駆動が行えるようにするべきです．今回設計した回路には用意していませんが，今後，この機能を追加する計画です．

● 乗算器の価値

基本論理ブロックの数だけを見ると，XC3S50は規模が小さすぎると感じるかもしれません．しかし，FPGAの中に乗算器があることを忘れてはいけません．

乗算器を使うと，例えば電子ボリュームを簡単に実現できます(**図9**)．電子ボリュームができれば，パンニングと呼ばれる，左右や前後に音がシフトする処理も実現できます(**図10**)．音の移動が実現できれば，映画「スターウォーズ」で見られたような，後方から戦闘機がレーザを発射して前方に移動するというシーンも表現できるのです．

パンニングについては，2チャネル・ステレオ出力を行うものとして，左右の音データにそれぞれ与えられたパラメータを掛けて足し合わせ，左右の音をミキシングします．パラメータの初期設定値は，左チャネルの出力用として左

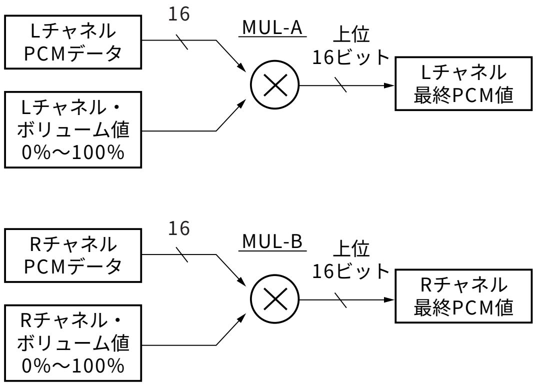
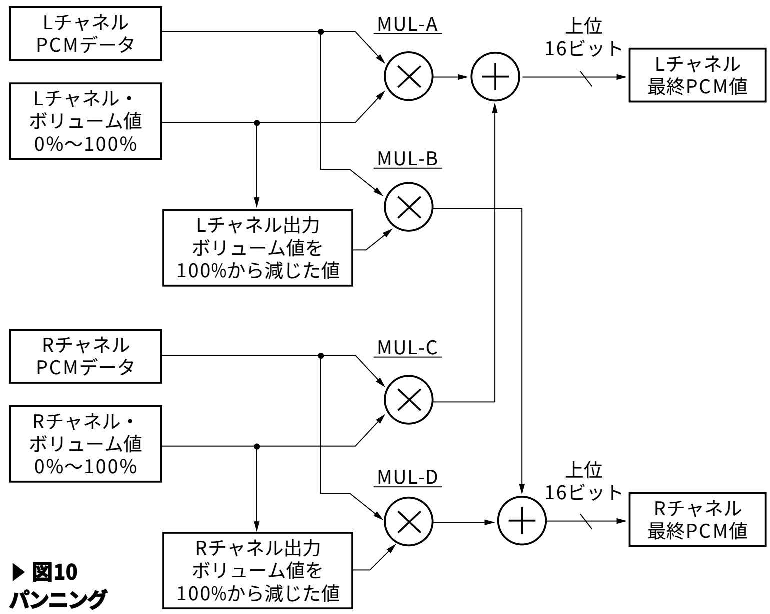


図9　電子ボリューム



▶図10　パンニング

側を100%とすることにします．結果として，左チャネルの音は左スピーカからしか聞こえません．ここでパラメータを70%とすると右側は30%となり，この30%分だけの音が右スピーカから聞こえます．逆の右チャネル出力も同様にパラメータ（右側を100%とする）を与えます．そうすると，右側にパラメータの音を乗じた出力が，左側には1－パラメータ%値を乗じた出力が出ます．これを左スピーカ出力，右スピーカ出力と互いに足し合わせれば，最終的な左右スピーカの出力になります．ただし，場合によっては片一方のチャネルだけに100%＋100%，合計200%のデータが来る場合がありますが，飽和を考えて1/2にする処理を行う必要はありません．というのは，人間の聴力はつごうの良いようにできていて，それなりに聞こえるからです．

このように電子ボリュームやパンニングは掛け算（乗算器）と足し算（加算器）を使うことで実現しますが，Spartan-3はこの乗算器を内部に実装しており，ユーザはVerilog HDL/VHDLのいずれからでもこのブロックを個別に指定して使用することができます．ISE WebPACK 6.3iのVHDL言語開発環境では，`C=A*B`と記述するだけで乗算器が使用されることを筆者は確認しています．つまり，HDLコードから乗算器が推定されます（**表8**）．

XC3S50では，PCMデータ・キュー・メモリの個数と同様に18ビット×18ビットの乗算器が4個入っており，これもまたちょうどいい形で音声処理に使えます．

すなわち，かりに電子ボリュームとして使用する場合の

表8　乗算器の評価

乗算器の使用	入力	出力（ビット）	ゲート数	LUT	SLICE	周波数（MHz）	乗算器使用数
OFF	18×18	36	4,772	315	175	76.7	0
OFF	16×16	32	3,833	248	140	92.1	0
OFF	18×8	24	1,925	116	64	117.3	0
OFF	8×8	16	1,061	60	36	126.4	0
ON	18×18	36	13,008	36	19	103.9	3
ON	18×17	35	8,767	17	9	132.4	2
ON	17×17	34	4,547	0	0	177.5	1
ON	16×16	32	4.515	0	0	181.8	1
ON	16×8	24	4,387	0	0	199.6	1
ON	8×8	16	4,259	0	0	225.9	1

（XC3S50-5VQ100＋ISE6.3を利用）

パラメータを8ビットとすると，16ビットのCDクオリティ・データ×8ビットの乗算器がどうしても必要となるのですが，この専用ブロックを使えば，内部ゲートや配線リソースを消費することなしに回路が実装できてしまうのです．

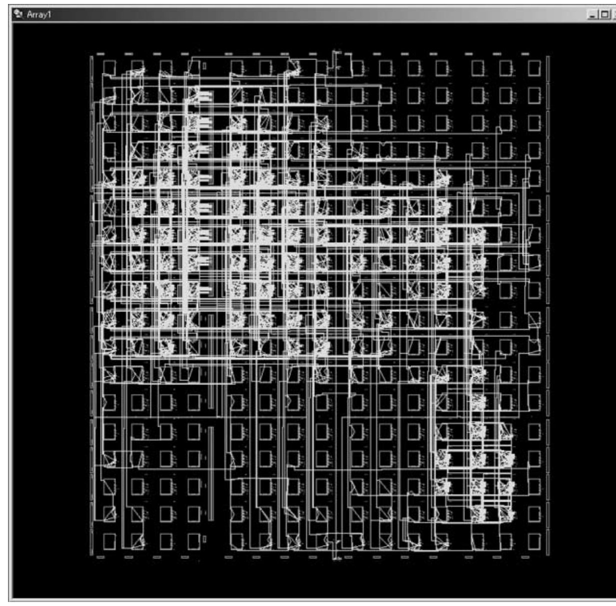
人間の音聴は優秀で，8ビット分解能（256段階）程度でも「急に音が変わったな」というのがわかってしまいます．やはり「実用的でここちよいサウンドを楽しむ」ための電子ボリュームとして使うには，10ビット分解能（1,024段階）～16ビット分解能（65,536段階）が必要です．ところが，これまではゲート規模の問題があり，8ビット分解能程度しか実装できませんでした．

一方，最近のFPGAは内部に専用の乗算器を備えています．これを利用することで，効率的に機能を実現することができます．今回はゴージャスに電子ボリュームの分解能

```
--------------Design Summary--------------
Number of errors : 0
Number of warnings : 0
Logic Utilization :
  Number of Slice Flip Flops :      173 out of   1,536     11%
  Number of 4 input LUTs :          311 out of   1,536     20%
Logic Distribution :
  Number of occupied Slices :       216 out of   768       28%
Total Number of 4 input LUTs :      311 out of   1,536     20%
  Number of bonded IOBs :           40 out of    63        63%
  IOB Flip Flops :                               49
  Number of Block RAMs :            2 out of     4         50%
  Number of GCLKs :                 2 out of     8         25%

Total equivalent gate count for design :   134,930
```

(**a**) 使用リソース



(**b**) 配置配線結果の状況

図11　FPGAへの実装結果

を16ビットに拡張し，16ビットCDクオリティのPCMデータに対して可変幅を持たせた電子ボリュームと24チャネル・ステレオ・パンニング機能を実現しました.

この回路自身は，一つの独立した回路モジュールです．AC-linkのPCMデータ・キューの読み出しとAC-linkシーケンサの入力回路の間にこれをはめ込むことで，実用に耐えうる回路が完成しました．設計結果を**図11**に示します．

4. まとめ

AC-linkの通信は，例えば256ビット長のシフト・レジスタを用意して，そこに毎回PCMデータやアクセス・レジスタの情報を書き込みシフト・アウトすることでも実現できます．しかし，48kHzごとにPCM音声データをかならずロードできるという条件を満足しない限り，この回路に実用性はありません．そこで，なんらかのバッファ回路が必要になります．

データをバッファリングすると，2倍速やスロー再生の機能を実現できます．細かな補間作業（通常はFIRフィルタを使用）を行えば，カラオケの音程変換などに使用される「サンプル・レート・コンバータ」という回路になります．また，ハードウェアによる電子ボリュームやパンニング，ミュート（音の出力を抑制する）操作を実現するためには，乗算器と加算器を使います．

こう考えると，音声処理には，「論理回路」＋「そこそこのメモリ」＋「乗算/加算ブロック」が必要ということになります．そして，コストも考慮すると，Spartan-3のようなFPGAが向いていると筆者は考えています．

AC-link制御は，CPUコアを使って実現することもできます．しかし，「なんでもかんでもCPUコア」ではおもしろくありません．回路規模も大きくなってしまいます．そこで，今回はあえてHDLにこだわって設計してみました．

付属FPGAの仕様を見て，「ゲート規模が小さくて，使いにくいのでは？」と思われた方もいるかもしれません．もちろん，ゲート規模が多いに越したことはないのですが，内部リソースをうまく使うことを考えれば，ゲート規模以上の回路を実現できます．

AC-linkコントローラを設計してみて，今回のFPGAが仕様からは見えないところで「小さくても小気味のいい使い勝手のFPGA」であることがわかりました．現在は，自作した回路をSH4-CPUカードに接続し，MP3データをデコードして音を聞いては満足感を味わっています．

参考・引用*文献

(1)* I²S bus specification, Philips Semiconductors, Feb, 1986.
(2)* Audio Codec' 97, Revision 2.3 Revision 1.0, Intel, April 2002.
(3)*WN9711L Low Power Audio CODEC for Portable Applications, Wolfson Microelectronics, April 2004.

いくら・まさみ
来栖川電工（有）
FPGAコンソーシアム会員